LA-UR-13-23338
Approved for public release; distribution is unlimited.

Title: MCNP5 to MCNP6 Migration Notes

Author(s): Goorley, John T.

Intended for: Release with MCNP6 via RSICC distribution. Also make this document available on mcnp.lanl.gov website

Issued: 2013-05-08

Disclaimer:
Los Alamos National Laboratory, an affirmative action/equal opportunity employer,is operated by the Los Alamos National Security, LLC for the National NuclearSecurity Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Departmentof Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

MCNP5 to MCNP6 Migrations Notes

Tim Goorley
4/25/2013

This document outlines the issues for users who are accustomed to MCNP5 and are interested in using MCNP6. MCNP6 operates (same style user interface and command-line execution) like previous versions of MCNP5, but users need to be aware of some minor differences. MCNP6 input files may require a few slight changes for them to work as would have been expected in MCNP5. This document is specific to the first MCNP6 production release, MCNP6.1.

I. Running.

Windows 64bit plotting now supported. MCNP5 did not support a 64 bit executable on Windows operating systems. MCNP6 includes the X11_64.dll for tally and geometry plotting on Windows.

Parallelism. MCNP6 precompiled executables have been built with the shared-memory threading (OMP), just like MCNP5. The expected mctal and output file templates are based on a OMP compiled executable. All MCNP5 MPI parallel compatible features are also MPI compatible in MCNP6. Not all MCNP6 features (mostly limited to the high-energy interaction models, but also ptrac and event logging) are compatible with OMP, and those are automatically detected and forced to run without threading. Threading can be forced on by setting dbcn(48) to a nonzero value.

Speed. MCNP6 is in most (but not all) cases slightly slower than MCNP5 by 5-20%.

New data path directories & default xsdir names

The naming convention for the xsdir filename has changed. The file xsdir_mcnp6.1 is the default searched for first by MCNP6. If this file is not found, then xsdir is searched for. The file xsdir is the default searched for first in MCNP5. Also note that xsdir_mcnp6.1_endfb-7.0 and xsdir_mcnp6.1_endfb-7.1were also created to allow users to easily select which set of nuclear data they wish to use by default (i.e. when no extension is listed on their ZAIDS, and no nlib keyword is used). A similar set has also been created for MCNP5. The MCNP6 version contains references to eprlib12 (which is incompatible with MCNP5 and MCNPX), while the MCNP5 version does not.

Change in Thermal neutron scattering S(a,b) data

With MCNP5, the thermal neutron scatting data library specified on the MT card (e.g. lwtr.10t for Hydrogen in light water) scattered neutrons in discrete angles. This resulted in a few “ray effects”. The default in MCNP6 (when the full ZAID extension is not used) is to use the continuous scattering available in ENDF/B-VII.1

II. Changes in input file defaults.

TOTNU is ON by default in MCNP6 for both KCODE and non-KCODE. The default in MCNP5 was OFF.

Particle designations. The particle designators (n,p,e) are the same in MCNP6 as they are in MCNP5. This particular list refers to neutrons, photons, and electrons, respectively. To specify a source particle type in the "PAR = " entry on an SDEF card, MCNP5 allowed either these letter designators or the corresponding numerical designators (1,2,3). This is still true in MCNP6. As a special syntax, only for the SDEF card, MCNP5 also allowed either the letter "f" or the number "4" to represent a positron source rather than an electron source. In MCNP6 the letter "f" continues to work for this specification (along with new possibilities "-e" or "-3") but the number "4" no longer refers to a positron. In MCNP6, specifying "4" on the PAR keyword on the SDEF card would result in a negative muon, not a positron. Note that the use of "f" to refer to a positron is still only a special syntax for the SDEF card, and cannot be used in other contexts such as IMP: or F: cards.

PHYS:N Some phys:n card entries have been moved to accommodate differences between MCNP5 and MCNPX. The fission multiplicity setting on the 5th entry of the phys:n card has been moved to the FMULT card. An entry made in 5th PHYS:N location is a fatal error in MCNP6v1. The following table describes how the older settings are mapped to the new input format.

Table 1. FMULT Settings in MCNP6

MCNP5 phys:n 5th entry	Description	MCNP6 FMULT card keyword setting
<0	Negative of Gaussian width for sampling multiplicities for all isotopes	Width=#
0	Bounded integer fission sampling	method=0, shift=0, data=0 [DEFAULT]
1	Lestone re-evaluated Gaussian width by isotope for multiplicities; neutrons resulting from an individual fission varies from 0 to approximately 10.	method=0, shift=0, data=1
2	Original Terrell Gaussian widths by isotope for multiplicities; neutrons resulting from an individual fission varies from 0 to approximately 10.	method=0, shift=0, data=2

There are several other methods (such as those based on Ensslin/Santi/Beddingfield/Mayo and the LLNL fission model) that are available in MCNP6 that aren’t available in MCNP5.

The specific method, shift and data parameter combinations listed here are the only ones assured to work correctly. Other combinations (such as method=1, shift=1 data=2) are possible but have not been tested. If a keyword is not specified, it is assumed to be 0.

Error Checking for Lattices. In MCNP5 and MCNPX, the following cell card entry was allowed for a cell which was filled with a lattice, but all those lattice locations were filled by the material and density (the 2nd and 3rd entries) of that cell:

```
7  3 -2.7  -11  12 -13  14 -15  16        u=2 lat=1 imp:n=1
```

In effect, the fill card used by the lattice was provided internally to MCNP and given the value of 0. While this is a reasonable intermediate state for constructing geometries, this would occasionally lead to lost particles and the associated geometry error was difficult to debug, in part because the plotter would not indicate this error with red dashed lines. Alternatively, this would also allow a convienient way to get tally results in each of the lattice location, but that function is now much easier (and able to be plotted) with mesh tallies (FMESH). In MCNP6 this lack of the fill card is not allowed, and results in the error message:

```
 fatal error.  Lattice in cell        7 has no fill.
```

For these cases, the user needs to add an additional cell with the appropriate material and density, and an unused universe number and the corresponding fill. In this example, the above cell should be modified to:

```
7   0      -11  12 -13  14 -15  16 fill=5 u=2 lat=1 imp:n=1
8  3 -2.7  -18                            u=5       imp:n=1
```

In the cases where the fill card is used to specify a range and then a finite number of filling locations, the zero fill entry is still valid. For example:

```
5  3 -1.0 -15  2 -18 17       imp:n=1 lat=1 u=2
           fill=0:1 0:3 0:0  4 4 4 4 0 5 4 4  $ 0 here is valid
```

Is a valid cell card description.

Use of data library driven transport. In MCNP5, nuclear data libraries had to be associated and found in the xsdir for each isotope present in their transport problem or a fatal error would result. In MCNP6 by default, users running neutrons, photons or electrons (or any subset thereof) also must have associated nuclear data libraries or a fatal error will result. This will protect users from a single character typo in their material cards listing ZAIDs.

However, by explicitly turning on physics models or by adding additional particle types to the mode card, no isotope is required to have any nuclear data associated with it at any energy range. Thus it is possible to have a single character typo in the ZAIDs and unintentionally use model based neutron-nuclear interaction cross sections in the MeV energy range (where models completely fail to predict threshold reaction cross sections). Warning messages will point this out.

DBCN(29)=1/0 (merger switch). With the release of MCNP6v1, there is no longer a flag to tell MCNP6 to run in either MCNP5 or MCNPX mode; both capabilities are always available.

This dbcn option has no effect on transport of particles. Please note that not all combinations of MCNPX capabilities and MCNP5 capabilities have been fully tested.

III. Documentation.

A user’s manual for MCNP6 is available, which focuses on the input card descriptions and example problems. It may be necessary to consult the MCNP5 and MCNPX manuals, especially the MCNP5 Manual Volume I – Theory, in combination with other documents associated with this release (most importantly the Production Release Overview and the Known Issues documents) for a more complete understanding of the code.

IV. Installing/Compiling.

New Installers. MCNP6 has new installation scripts. The most significant difference is that the MCNP6 Windows installers no longer use InstallShield (to copy and install files) nor 7z.exe (to uncompress files).

Build system. MCNP6 follows the MCNP5 method for configuring and compiling the code (e.g. make build CONFIG=’intel plot omp’), see the README file in the top-level source directory. You must have Cygwin installed (specifically the make utility and perl) to build on Windows machines. MCNP6 supports a different set (upgraded) of Fortran compilers than MCNP5 does. The recommend choice is still Intel Fortran, but now a more recent version (Intel 12.1.5). Note: as of the current release, MCNP6 will not compile with any version of Intel Fortran 13.